

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-320184

(43)Date of publication of application : 04.12.1998

(51)Int.Cl.

G06F 9/06
G06F 9/445
G06F 13/00
G06F 13/00

(21)Application number : 09-147112

(71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 21.05.1997

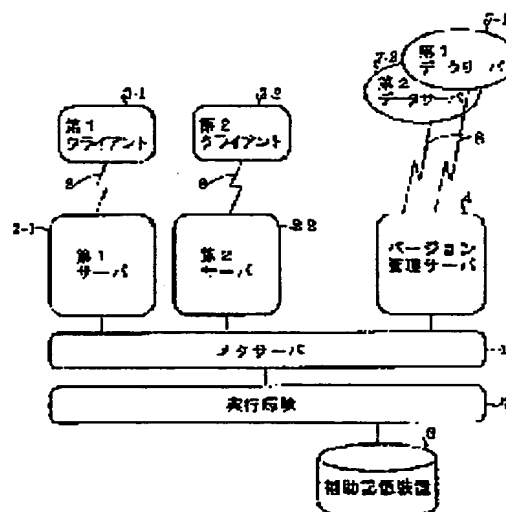
(72)Inventor : FUKUDA HARUO

(54) SOFTWARE VERSION MANAGEMENT SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To automatically revise a server by providing a metaserver which provides environment where the server operates and a version management server which manages the version of programs on the server by accepting information at a program provision source.

SOLUTION: For example, when a corresponding program on a 2nd data server 7-2 is updated, the version management server 4 downloads the corresponding program. The version management server 4 sends a service stop request to corresponding servers 2-1 and 2-2. The servers 2-1 and 2-2 having accepted the service stop request closes the ports of a network. The version management server 4 loads a loaded program in execution environment 5 and starts up a new server having been revised. The metaserver starts the new server. The new server starts providing services. The version management server 4 deletes the old servers from the execution environment 5.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-320184

(43) 公開日 平成10年(1998)12月4日

(51) Int.Cl.⁶
G 0 6 F 9/06 4 1 0
9/445
13/00 3 5 1
3 5 7

F I
G 0 6 F 9/06 4 1 0 P
13/00 3 5 1 H
3 5 7 Z
9/06 4 2 0 J

審査請求 未請求 請求項の数 4 F D (全 7 頁)

(21) 出願番号 特願平9-147112
(22) 出願日 平成9年(1997)5月21日

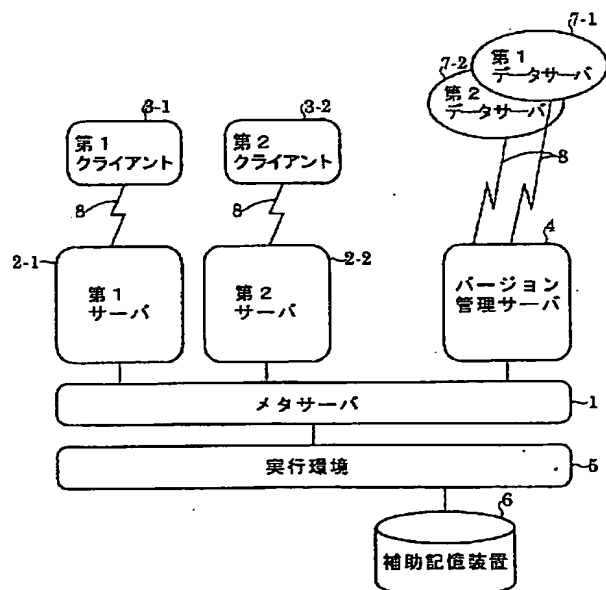
(71) 出願人 000000295
沖電気工業株式会社
東京都港区虎ノ門1丁目7番12号
(72) 発明者 福田 春生
東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内
(74) 代理人 弁理士 佐藤 幸男 (外1名)

(54) 【発明の名称】 ソフトウェアバージョン管理システム

(57) 【要約】

【解決手段】 第1サーバ2-1や第2サーバ2-2は、クライアントにサービスを提供する。メタサーバ1はこれらのサーバのプログラムを起動させたり停止させる制御を行う。バージョン管理サーバ4はネットワーク8を通じてデータサーバから各サーバ2-1や2-2のプログラムをダウンロードし、所定のタイミングで自動的に各サーバのプログラムをバージョンアップする。

【効果】 バージョン管理サーバ4がバージョンアップの作業を管理し、メタサーバ1が自動的に各サーバのプログラムの起動や停止を制御するので、バージョンアップ処理を自動化できる。



具体例1によるソフトウェア管理システムブロック図

【特許請求の範囲】

【請求項 1】 ネットワークに接続されて、クライアントにサービスを提供するサーバと、このサーバが動作するための環境を提供するメタサーバと、

ネットワークを通じてプログラム提供元の情報を受入れて、前記サーバのプログラムのバージョンを管理するバージョン管理サーバとを備え、

サービスの開始時には、前記メタサーバが各サーバのプログラムを起動し、

バージョン管理サーバが、ネットワークを通じて前記プログラム提供元の情報を受入れて、各サーバで動作するプログラムのバージョン変更を認知すると、最新のプログラムをプログラム提供元からダウンロードし、

メタサーバが、該当するサーバのプログラムを停止し、バージョン管理サーバは既にダウンロードした最新のプログラムをロードして、

メタサーバがそのサーバのプログラムを再起動することを特徴とするソフトウェアバージョン管理システム。

【請求項 2】 請求項 1 において、

バージョン管理サーバは、

バージョン変更前のサーバの動作を継続させたまま、新たなバージョンのサーバをロードし、

新たなバージョンのサーバの起動準備が完了したとき、クライアントに対して、バージョン変更前のサーバを接続した状態から新たなサーバに接続をした状態に切り替えるためのネットワークブリッジを設けたことを特徴とするソフトウェアバージョン管理システム。

【請求項 3】 クライアントにサービスを提供するサーバをネットワークに接続し、

前記サーバの動作するプログラムについてバージョンを管理するバージョン管理サーバを設け、

サービスの開始時には、各サーバのプログラムを起動し、

バージョン管理サーバが、ネットワークを通じて前記プログラム提供元の情報を受入れて、各サーバで動作するプログラムのバージョン変更を認知すると、最新のプログラムをプログラム提供元からダウンロードし、

該当するサーバのプログラムを停止し、

既にダウンロードした最新のプログラムをロードして、プログラムを再起動するように動作するプログラムを記録した記録媒体。

【請求項 4】 請求項 3 において、

バージョン管理サーバは、

バージョン変更前のサーバの動作を継続させたまま、新たなバージョンのサーバをロードし、

新たなバージョンのサーバの起動準備が完了したとき、クライアントに対して、バージョン変更前のサーバを接続した状態から新たなサーバに接続をした状態に切り替えるよう動作するプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、サーバクライアントシステムを構成するサーバの動作プログラムを自動的にバージョンアップする機能を持つソフトウェア管理システムに関する。

【0002】

【従来の技術】例えば、ウェブブラウザ上で動作するジャバ言語により作成されたプログラム（アプレット）

は、起動時に自動的にネットワークを経由してプログラムが転送される仕組みを持っている。このため、アプレット使用時には常に最新のプログラムが利用できるという利点がある。即ち、従来一般のアプリケーションソフトウェアのように、バージョンアップの度にユーザがネットワークやその他のメディアを通じて最新版を入手し、自らバージョンアップ作業を行う必要がない。

【0003】しかしながら、アプレットは毎回起動時にプログラム全体をダウンロードすることから、ネットワーク負荷が高くなる。また、起動開始から起動までにダウンロードによるタイムラグが発生する。

【0004】一方、アプレットを通常のアプリケーションのように各端末の持つハードディスク上に保存しておく、ダウンロード元のサーバ上にあるファイルが更新されたときに限り、必要最小限のファイルをネットワークによりダウンロードする方法もある。これによれば、最初の起動時以外はローカルなハードディスク上にファイルが存在するため、高速起動が可能になる。また、更新時のファイルのダウンロードは必要最小限のものに限られるので、ネットワーク負荷を高めることはない。

【0005】

【発明が解決しようとする課題】ところで、上記のような従来の技術には次のような解決すべき課題があった。上記のようなバージョンアップ方法は、ワードプロセッサや各種の管理端末装置等で動作するアプリケーションを対象としている。従って、これらのアプリケーションのバージョンアップはそのアプリケーションを実行する直前等に十分な時間的余裕を持って処理することができる。

【0006】しかしながら、サーバクライアントシステムのサーバが動作するプログラムは、これと同様の方法を採用すると弊害が生じる。即ち、サーバのプログラムは常時動作しており、クライアントからの要求に応じて所定のサービスを提供している。従って、管理者によるバージョンアップ作業のために長時間サーバの動作を停止させることができない。このようなサービスの停止を避けるために、サーバの利用が少ない深夜等にバージョンアップ作業を行おうとすれば、管理者に深夜作業を強いることになる。

【0007】

【課題を解決するための手段】本発明は以上の点を解決

するため次の構成を採用する。

〈構成1〉ネットワークに接続されて、クライアントにサービスを提供するサーバと、このサーバが動作するための環境を提供するメタサーバと、ネットワークを通じてプログラム提供元の情報を受入れて、上記サーバのプログラムのバージョンを管理するバージョン管理サーバとを備え、サービスの開始時には、上記メタサーバが各サーバのプログラムを起動し、バージョン管理サーバが、ネットワークを通じて上記プログラム提供元の情報を受入れて、各サーバで動作するプログラムのバージョン変更を認知すると、最新のプログラムをプログラム提供元からダウンロードし、メタサーバが、該当するサーバのプログラムを停止し、バージョン管理サーバは既にダウンロードした最新のプログラムをロードして、メタサーバがそのサーバのプログラムを再起動することを特徴とするソフトウェアバージョン管理システム。

【0008】〈構成2〉構成1において、バージョン管理サーバは、バージョン変更前のサーバの動作を継続させたまま、新たなバージョンのサーバをロードし、新たなバージョンのサーバの起動準備が完了したとき、クライアントに対して、バージョン変更前のサーバを接続した状態から新たなサーバに接続をした状態に切り替えるためのネットワークブリッジを設けたことを特徴とするソフトウェアバージョン管理システム。

【0009】〈構成3〉クライアントにサービスを提供するサーバをネットワークに接続し、上記サーバの動作するプログラムについてバージョンを管理するバージョン管理サーバを設け、サービスの開始時には、各サーバのプログラムを起動し、バージョン管理サーバが、ネットワークを通じて上記プログラム提供元の情報を受入れて、各サーバで動作するプログラムのバージョン変更を認知すると、最新のプログラムをプログラム提供元からダウンロードし、該当するサーバのプログラムを停止し、既にダウンロードした最新のプログラムをロードして、プログラムを再起動するように動作するプログラムを記録した記録媒体。

【0010】〈構成4〉構成3において、バージョン管理サーバは、バージョン変更前のサーバの動作を継続させたまま、新たなバージョンのサーバをロードし、新たなバージョンのサーバの起動準備が完了したとき、クライアントに対して、バージョン変更前のサーバを接続した状態から新たなサーバに接続をした状態に切り替えるよう動作するプログラムを記録した記録媒体。

【0011】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

〈具体例1〉図1は、具体例1によるソフトウェア管理システムブロック図である。図のシステムは、メタサーバ1により起動や停止を制御される。第1サーバ2-1や第2サーバ2-2は、第1クライアント3-1や第2

クライアント3-2に対し所定のサービスを提供するサーバである。また、本発明による第1サーバ2-1や第2サーバ2-2のバージョンアップ管理のために、メタサーバ1にはバージョン管理サーバ4が接続されている。

【0012】メタサーバ1は、この動作環境で第1サーバ2-1や第2サーバ2-2が動作するための共通の環境を提供するサーバである。第1サーバ2-1や第2サーバ2-2は、DNSウェブサーバ等どんなサーバであってもよく、その数は任意である。メタサーバ1に接続された実行環境5は、OS（オペレーションシステム）等のソフトウェアを動作させるために必要な実行環境であって、ジャバ言語の仮想マシン等も含まれる。

【0013】バージョン管理サーバ4は、メタサーバ1上で動作している全てのサーバに関する情報を管理するサーバである。バージョン管理サーバ4も任意の数だけ設けられていてよい。補助記憶装置6は、実行環境5を介してメタサーバ1に接続されている。この補助記憶装置6は、バージョン管理サーバ4や第1サーバ2-1、第2サーバ2-2等がメタサーバ1と実行環境5を介してアクセスするための記憶装置である。

【0014】なお、各サーバがメタサーバ1を介すことなく独自の補助記憶装置を持つようにしても差し支えない。第1サーバ2-1や第2サーバ2-2は、ネットワーク8を介して第1クライアント3-1や第2クライアント3-2に接続されている。一方、バージョン管理サーバ4もネットワーク8を介して、第1データサーバ7-1や第2データサーバ7-2に接続されている。

【0015】第1データサーバ7-1は、例えば、メタサーバ1やバージョン管理サーバ4の動作プログラムの最新版を格納し公開している。第2データサーバ7-2は、第1サーバ2-1や第2サーバ2-2の動作する最新のプログラムを格納し公開している。なお、各サーバのためのプログラムごとにそれぞれ異なるデータサーバが存在していても構わないし、1台のデータサーバがこれら全てのプログラムを公開するようにしても差し支えない。

【0016】第1サーバ2-1、第2サーバ2-2、メタサーバ1、バージョン管理サーバ4等は、1台のコンピュータメモリ上に展開されたプログラム群により構成してもよいし、またそれぞれ相互に通信用のバスで接続された複数のプロセッサにより構成してもよい。

【0017】次に、上記のシステムについて、その動作を説明する。システムの動作は基本的に、サーバの初期設定、サーバの運用、サーバのバージョン変更の3種類に分類できる。

【0018】〈サーバの初期設定〉図2に、サーバの初期設定動作フローチャートを示す。まず、サーバを利用するために、図のステップS1とS2に示すように、メタサーバ1とバージョン管理サーバ4のインストールを

行う。これらのプログラムは、図 1 に示すネットワーク 8 を介して第 1 データサーバ 7-1 から入手する。そして、これらのプログラムをサーバとなるマシンにインストールし、ステップ S 3 において、メタサーバとバージョン管理サーバを起動する。

【0019】次に、ステップ S 4 において、バージョン管理サーバ 4 に対し、新たにサーバとなる第 1 サーバと第 2 サーバの登録を行う。例えば第 1 サーバ 2-1 をバージョン管理サーバ 4 に登録する場合は、このサーバの最新版となるプログラムを提供する第 2 データサーバ 7-2 のネットワーク上のアドレスと、利用するサーバのプログラム名を通知することで実現する。

【0020】次のステップ S 5 では、バージョン管理サーバ 4 が第 2 データサーバ 7-2 からネットワーク 8 を通じて第 1 サーバと第 2 サーバのプログラムをダウンロードする。このとき、第 1 サーバや第 2 サーバの動作に必要なデータファイルがある場合には同時にダウンロードをする。なお、このときのネットワークプロトコルには、従来からよく知られている HTTP（ハイパーテキスト転送プロトコル）や FTP（ファイル転送プロトコル）を利用することができる。

【0021】HTTP であれば、プロキシキャッシュ等の機構を利用することで効率的にプログラムのダウンロードを行うことができる。なお、第 2 データサーバ 7-2 の側でも、バージョン管理サーバ 4 が管理しているマシンを登録しておく。これは第 2 データサーバ 7-2 の側からプログラムのバージョンアップをバージョン管理サーバ 4 に自発的に通知できるようにするためである。

【0022】なお、ここまででサーバの初期設定が終了し、ステップ S 6 とステップ S 7 では、サーバの運用が開始される。即ち、ステップ S 6 において、第 1 サーバ 2-1 と第 2 サーバ 2-2 とは、メタサーバ 1 によってそのプログラムが起動される。そして、ステップ S 7 において、第 1 サーバと第 2 サーバとは通常のサービス提供を開始する。これによって、図 1 に示した第 1 クライアント 3-1 や第 2 クライアント 3-2 は、ネットワーク 8 を通じて第 1 サーバ 2-1 や第 2 サーバ 2-2 のサービスを受けることができる。

【0023】〈サーバのバージョン変更〉図 3 は、バージョン変更動作フローチャートを示す。運用が開始されると、バージョン管理サーバ 4 は図のステップ S 1 に示すように、第 1 サーバや第 2 サーバのプログラム更新を監視する。これは、例えばネットワーク 8 を通じて第 2 データサーバ 7-2 に対し定期的に問い合わせを行うか、あるいは第 2 データサーバ 7-2 からの通知を受けることにより実現する。第 2 データサーバ 7-2 上の該当するプログラムが更新されると、バージョン管理サーバ 4 はステップ S 2 からステップ S 3 に進み、該当するプログラムをダウンロードする。

【0024】次にステップ S 4 において、バージョン管

理サーバ 4 は該当するサーバに対してサービス停止要求を行う。なお、これらの一連の処理は全て自動的に行われるため、サービスの停止要求は深夜等、サーバに対する需要が少ない時間を任意に設定することが好ましい。第 1 サーバ 2-1 や第 2 サーバ 2-2 はステップ S 5 において、サービス停止要求を受けると、ネットワークのポートをクローズする。こうしてクライアントからの要求の受付を停止する。次に、ステップ S 6 において、その内部状態を、図 1 に示した補助記憶装置 6 に出力する。これは、バージョンアップ後、現在の設定から起動できるようにしておくためである。

【0025】こうして停止処理を終えた旧バージョンのサーバは、停止処理終了をバージョン管理サーバに通知する。バージョン管理サーバ 4 は次のステップ S 8 において、ステップ S 3 でダウンロードしたプログラムを実行環境中にロードし、バージョン変更後の新サーバを立ち上げる。そして、ステップ S 9 において、メタサーバ 1 は新サーバを起動する。次に、ステップ S 10 で、新サーバは図 1 に示した補助記憶装置 6 から保存しておいた内部状態を引き上げて、サービスを再開する。その後、ステップ S 11 において、バージョン管理サーバ 4 は、旧サーバを実行環境から削除する。

【0026】図 4 には、上記のような具体例 1 のシステムの動作を主要なブロックと共に図示した。ここには、メタサーバ 1、バージョン管理サーバ 4、補助記憶装置 6、旧サーバ 2 及び新サーバ 2' のみが表示されている。図 3 に示した各ステップは、この図の矢印に対応する。即ち、バージョン管理サーバ 4 は、ステップ S 4 で旧サーバ 2 を停止し、旧サーバ 2 はステップ S 6 で、内部状態を補助記憶装置 6 に記憶する。ステップ S 7 で、旧サーバ 2 がバージョン管理サーバ 4 に停止を通知すると、ステップ S 8 で、バージョン管理サーバ 4 は新サーバ 2' を立ち上げる。

【0027】新サーバ 2' はステップ S 10 で補助記憶装置 6 から内部状態を引き上げて、動作を開始し、その後、バージョン管理サーバがステップ S 11 で旧サーバ 2 をシステムから除去する。このように旧サーバ 2 から新サーバ 2' に切換えを行う処理は自動的に行われる。また、バージョン管理サーバ 4 がネットワークからプログラムをダウンロードした後に、コンピュータのメモリ上で速やかに実行される。

【0028】〈具体例 1 の効果〉以上の具体例 1 によれば、メタサーバ 1 とバージョン管理サーバ 4 とを設けることによって、サーバのバージョンアップ処理を自動的に行うことが可能になる。また、こうしたバージョンアップは自動的に一定の手順で実行されるため、管理者が手作業で行う場合よりも迅速に処理でき、サーバの停止時間が短時間で済む。さらに、この作業は自動化されるため、夜中等にも実施が可能である。なお、こうした処理は、ネットワークに既存のプロトコルが使用できるの

で、新たなシステムを構築する必要がないという効果もある。

【0029】〈具体例2〉図5には、具体例2によるソフトウェア管理システムのブロック図を示す。このシステムは、図1のシステムと比較してわかるように、メタサーバ1上で動作する各サーバが、ネットワーク8の側において、ネットワークブリッジ9によって連結されている。従って、この構成では、第1クライアント3-1はネットワークブリッジ9を介して第1サーバ2-1からサービスを受けることもできるし、第2サーバ2-2からサービスを受けることもできる。

【0030】また、第2クライアント3-2も同様に、ネットワークブリッジ9を介して第1サーバ2-1からサービスを受け、あるいは第2サーバ2-2からサービスを受けることができる。このようにネットワーク接続を仲介するネットワークブリッジ9を設けることによって、この具体例ではバージョンアップの際に、サーバの動作を停止することなくその処理を完了することができる。

【0031】具体例2の場合のサーバの初期設定や運用動作等は、具体例1の場合と全く同様のため説明を省略する。また、バージョンアップについても、その多くの部分は具体例1と同様のため、フローチャートは省略し、次の動作説明図によって主たる相違部分を説明する。

【0032】図6は、具体例2のシステムの動作説明図である。図4と比較してわかるように、具体例1では、一旦旧サーバの動作を停止し、内部状態を補助記憶装置6に出力し、その後新サーバ2'を立ち上げてバージョン変更を行った。一方、この具体例では、ネットワークブリッジ9を設けるとともに、旧サーバ2のサービス停止を新サーバ2'のサービス開始に合わせるようにした。これによって、バージョン変更処理の実行中にサーバが停止するのを防止する。

【0033】図6の矢印に書き入れたステップS1～S7までの処理を参照しながら、この切換え動作を具体的に説明する。まずステップS1において、バージョン管理サーバ4は旧サーバ2に対しサービス停止の要求を行う。旧サーバ2はステップS2において、補助記憶装置6に内部状態を出力する。なお、旧サーバ2はこの時点ではサービスを停止せず、サービスを継続する。そして、ステップS3において、一定の処理が終了したことをバージョン管理サーバ4に通知する。

【0034】バージョン管理サーバ4はこの通知を受けて、ステップS4において、新サーバ2'をロードし、立上げを行う。新サーバ2'は、補助記憶装置6から旧サーバの内部状態を取り出し、サービス開始準備を行

う。そして、準備が完了するとステップS6でバージョン管理サーバ4にその旨を通知する。バージョン管理サーバ4は、新サーバ2'からサービス提供準備完了の通知を受けると、ステップS7において、ネットワークブリッジ9に対しサービス接続先の変更を通知する。即ち、ネットワークブリッジ9は、これまで旧サーバ2とネットワーク8とを接続し、第1クライアント3-1に対するサービス提供を行ってきた。

【0035】バージョンアップの準備が完了すると、今度はネットワークブリッジ9は新サーバ2'とネットワーク8を接続し、旧サーバ2をネットワーク8から切り離す。こうしてその後は新しいサービスが新サーバ2'からクライアント3に提供される。こうしてネットワーク8から切り離された旧サーバ2は、ステップS8でバージョン管理サーバ4から動作停止要求を受けて動作を停止する。そして、ステップS9で旧サーバ2は動作の停止をバージョン管理サーバ4に通知する。その後は具体例1と同様に、バージョン管理サーバ4は旧サーバ2を実行環境から削除してバージョンアップ処理を終了する。

【0036】〈具体例2の効果〉以上のように、バージョン変更の際、旧サーバと新サーバとをネットワークブリッジを介してクライアントに接続し、バージョンを変更した新サーバがサービス提供開始準備のできるまで、旧サーバにサービスを継続させるので、バージョン変更処理によってサービスが停止することがない。

【図面の簡単な説明】

【図1】具体例1によるソフトウェア管理システムのブロック図である。

【図2】サーバの初期設定動作フローチャートである。

【図3】バージョン変更動作フローチャートである。

【図4】具体例1のシステムの動作説明図である。

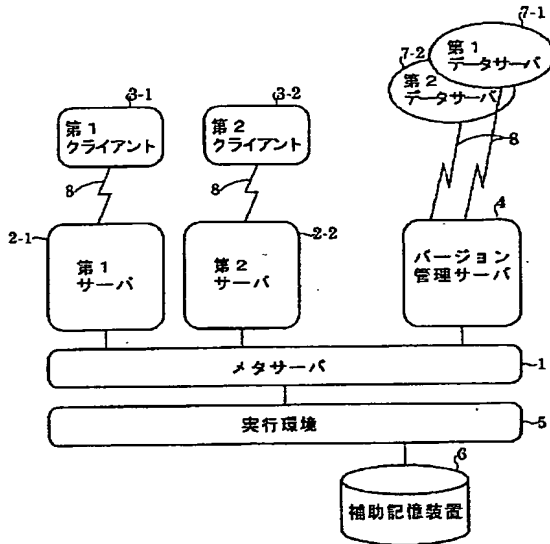
【図5】具体例2によるソフトウェア管理システムブロック図である。

【図6】具体例2のシステムの動作説明図である。

【符号の説明】

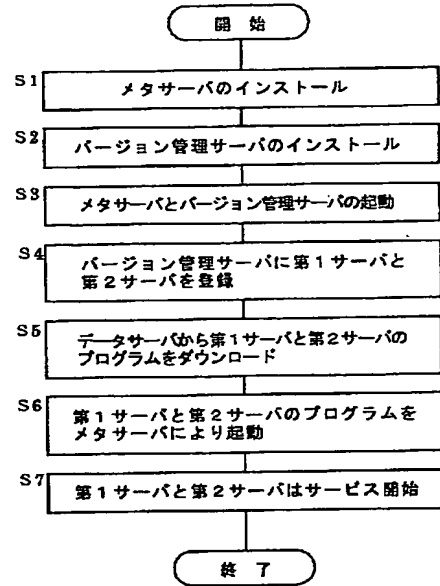
- 1 メタサーバ
- 2-1 第1サーバ
- 2-2 第2サーバ
- 3-1 第1クライアント
- 3-2 第2クライアント
- 4 バージョン管理サーバ
- 5 実行環境
- 6 補助記憶装置
- 7-1 第1データサーバ
- 7-2 第2データサーバ
- 8 ネットワーク

【図1】



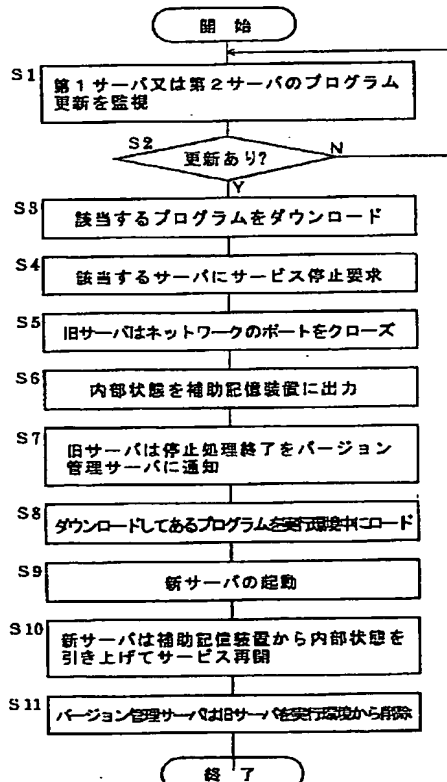
具体例1によるソフトウェア管理システムブロック図

【図2】



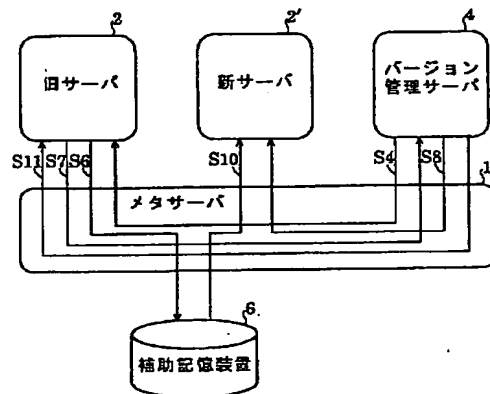
サーバの初期設定動作フローチャート

【図3】



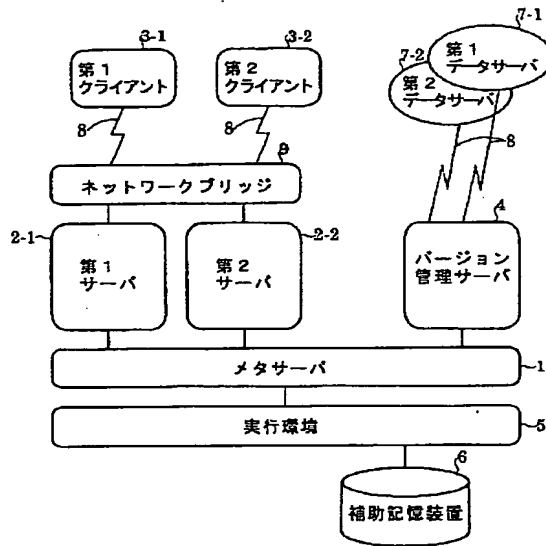
バージョン更新動作フローチャート

【図4】



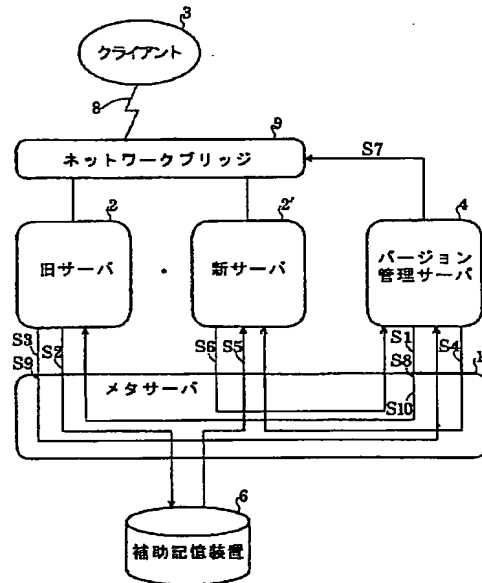
具体例1のシステムの動作説明図

【図5】



具体例2によるソフトウェア管理システムブロック図

【図6】



具体例1のシステムの動作説明図